

MSX verder uitgediept

screen locaties * sprites * control codes
vpoke en het scherm * de video chip
screen save * disk loader * header lezer
de vdp registers * cursor manipulatie

H. Klopper



verder uitgediept

H. Klopper

uitgeverij STARK-TEXEL

postbus 302 - 1794 ZG Oosterend tel. 02223 - 661

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Klopper, H.

MSX verder uitgediept. / H. Klopper. — Oosterend : Stark-Textel
ISBN 90-6398-447-2
SISO 365.3 UDC 681.3
Trefw.: MSX (computer).

1e druk 1986
ISBN 90 6398 447 2

© by uitgeverij Stark-Textel, Oosterend Nh.

Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

No part of this book may be reproduced in any form, by print, photo-print, microfilm or any other means without written permission from the publisher.

Ondanks alle aan de samenstelling van de tekst bestede zorg kan noch de redactie noch de uitgever aansprakelijkheid aanvaarden voor eventuele schade die zou kunnen voortvloeien uit enige fout die in deze uitgave zou kunnen voorkomen.

MSX is een handelsmerk van Microsoft

INHOUD

	pag.
Inleiding	7
Cursor manipulatie	11
Het veranderen van de karakterset	12
Het gebruik van VPOKE bij het scherm	15
De funktietoetsen	16
MSX Control Codes	17
(Video-)Geheugen indeling	19
De Video chip in uw MSX computer	22
Uitleg van de screen locaties	25
Sprites in mode 1, 2 en 3	30
De VDP registers	33
Het VDP statement	34
Het Base statement	40
Appendix A	
Het hexadecimale getallen systeem	41
Appendix B	
Binarie getallen	43
Programma's	
Disk loader utility	46
Animatie met karakters	49
Lijnen verdwijnen	50
Beeldscherm wegschrijven	52
Headerlezer **Disk utility**	53
Rom-routines	56
MSX Logo	63
Scroll	65
Simulatie opstartscherm	66
Willekeurige kleuren	68
Balspel	69
Wereldkaart	72

INLEIDING

MSX computers hebben een 32K byte groot ROM geheugen. Meestal hebben deze computers daarnaast een RAM geheugen van 80K bytes.

- Het ROM geheugen bevat programmatuur geschreven in machinecode. Deze programmatuur zorgt ervoor dat de door u ingevoerde instructies onder BASIC begrepen worden door de computer.
Machinetaalprogrammeurs maken vaak gebruik van de 'routines' in deze ROM, wat het gemak bij het schrijven van een machinetaalprogramma sterk vergroot.
- Het RAM geheugen bevat in principe geen programmatuur en is dan ook geschikt voor het invoeren van de door u te ontwerpen programmatuur.

Het ROM geheugen is alleen uit te lezen. In het RAM geheugen kunt u zowel informatie wegschrijven, als informatie uitlezen.

Doordat het MSX systeem is opgebouwd rondom de Z80A microprocessor kan er niet meer dan 64K aan geheugen tegelijkertijd gebruikt worden (dit geldt voor elke '8 bitter'). Een simpel rekensommetje ($80+32=112$) levert ons een getal op dat veel groter is dan deze 64K. Hoe zit dat nu?

Als u in BASIC programmeert is het ROM geheugen permanent in werking om ervoor te zorgen dat de computer u ook daadwerkelijk begrijpt - dat is dan al 32K - de overige 32K staat volledig tot uw beschikking in de vorm van RAM. Nu zult u denken wat gebeurt er nu met het overige 48K RAM geheugen dat nog in m'n MSX machine is gestopt?

16K wordt gebruikt voor de opbouw van het beeldscherm. Deze 16K wordt dan ook volledig bestuurd door de Video Chip. Hierdoor wordt bij gebruik van 16 kleuren in high resolution (screen 2) geen geheugen van de 32K RAM afgenomen (zoals bij andere computersystemen). Dit Video

RAM kan heel goed door u worden gebruikt, maar meer hierover later.

De overige 32K RAM kan door bovengenoemde beperking niet direct onder BASIC door u worden gebruikt. In machinetaal heeft u vrijwel het gehele RAM geheugen tot uw beschikking.

Overigens bestaan er voor de BASIC programmeur utilities waardoor het zogenaamde bankswitching mogelijk wordt. Bankswitching maakt het mogelijk om toch van de 32K extra RAM gebruik te maken.

Voor informatie hierover kunt u terecht bij uw MSX software dealer.

De BASIC van uw MSX computer is een van de meest krachtige BASIC versies die momenteel beschikbaar zijn op homecomputers. Vroeger was het voor een homecomputergebruiker vrijwel een verplichting om de computer bijna net zo goed te leren kennen als de ontwerper van het apparaat. Dit kwam doordat alle extra faciliteiten als grafiek en geluid niet werden ondersteund door de BASIC in deze machines. Peek en Poke waren de toverwoorden in die tijd.

Waarom dan een boek over Peek en Poke voor MSX computers?

De BASIC heeft weliswaar zeer veel mogelijkheden, maar voor zeer verfijnde besturing van uw computer is er vaak toch iets extra's nodig.

Hiervoor moet u dan toch de beruchte Peek- en Poke-instructies gebruiken.

MSX computers hebben de beschikking over een internationale karakter set. Zowel hoofdletters als kleine letters zijn beschikbaar.

Om te zien welke tekens er in uw MSX liggen opgeslagen kunt u programma nr.1 gebruiken:

```
10 SCREEN0:WIDTH32
20 FORX=0TO255
30 PRINTCHR$(X);
```

40 NEXT

De decimale waarden 32 t/m 126 komen overeen met de door ASCII gestandaardiseerde codes. Deze codes zijn op elke computer die gebruik maakt van ASCII codes gelijk.

Een andere manier om de MSX karakters op het scherm te krijgen demonstreert programma nr.2:

```
10 SCREEN0
20 FORX=0TO255
30 VPOKE X,X
40 NEXT
```

U ziet dat nu ook 'smiles', 'muzieknoten' etc. op het scherm komen, waarom? Dat zullen we later laten zien . . .

Het tekstschermb (screen 0) is verdeeld in maximaal 40 kolommen (vertikaal) en 24 rijen (horizontaal). De onderste rij wordt gebruikt voor het afbeelden van de functie-toetsinhoud.

Een van de beperkingen van de MSX karakterset, nl. geen inverse tekens, kunt u oplossen m.b.v. programma nr.3:

```
10 SCREEN0
20 FORI%=2304 TO 3055
30 A%=VPEEK(I%)
40 A%=A%XOR255
50 VPOKE I%+768,A%
60 NEXT
```

Tik nu in: `FOR X=1TO 255:PRINTCHR$(X);:NEXT`

U ziet dat er een inverse-karakterset achter de normale is 'gevpoked'. U kunt deze karakters op het scherm krijgen

d.m.v. het printen van de CHR\$-codes of door gebruik te maken van de CODE toets in combinatie met de letters op het toetsenbord.

CURSOR MANIPULATIE

U kunt de cursor uit zetten d.m.v. locate,,0 en weer aan d.m.v. locate,,1.

Dit geeft te denken. Een knipperende cursor bij een invoer-routine bijvoorbeeld, is mogelijk als u gebruik maakt van bovengenoemde instructies. Voor deze routinegebruikt u programma nr.4:

```
10 SCREEN0:WIDTH38
20 PRINT"druk een toets"
30 LOCATE 23,0,0
40 FORT=1T0100:NEXT:LOCATE,,0:A$=INKEY$
50 IFA$=""THEN100
60 END
100 LOCATE23,0,1:FORT=1T050:NEXT
110 GOTO 30
```

HET VERANDEREN VAN DE KARAKTERSET

De vorm van de karakters wordt na het inschakelen van de computer vanuit het ROM (start 1BBFH) overgebracht naar het Video RAM. De karakters zijn opgebouwd uit een 8 bij 8 matrix.

De locaties in het Video RAM zijn te zien met het corresponderende codenummer. Met programma nr.5 kunnen we de locaties onderzoeken, de nullen laten de lege plaatsen zien en de enen laten de vorm van de karakters zien.

Programma nr.5:

```
10 SCREEN0:WIDTH38
15 Y=2047:FORX=2048TO4096
20 S=VAL(BIN$(VPEEK(X)))
30 B$="#### #####"
40 PRINT USINGB$;X,S
50 IF X=Y+S THEN PRINTELSE GOTO70
60 Y=X
70 NEXT
```

P.S. Voor SCREEN mode 1:

```
10 SCREEN1:WIDTH30
15 Y=-1:FORX=0TO2047
```

Nu we weten waar de locaties van de karakters te vinden zijn, kunnen we met de VPOKE instructie nieuwe byte codes invoeren en zo de vorm van een bepaald karakter veranderen.

De nu volgende programma's nr.6a en 6b demonstreren het gebruik van de locaties om de vorm van de letter 'A' te veranderen.

Programma nr.6a:

```
10 SCREEN0:WIDTH38
20 VPOKE 2568,&B00011000
30 VPOKE 2569,&B00111100
40 VPOKE 2570,&B01111110
50 VPOKE 2571,&B00111100
60 VPOKE 2572,&B01111110
70 VPOKE 2573,&B11111111
80 VPOKE 2574,&B00011000
90 VPOKE 2575,&B00011000
100 PRINT"A"
```

Programma nr.6b:

```
10 SCREEN1:WIDTH30
20 VPOKE 520,&B00011000
30 VPOKE 521,&B00111100
40 VPOKE 522,&B01111110
50 VPOKE 523,&B00111100
60 VPOKE 524,&B01111110
70 VPOKE 525,&B11111111
80 VPOKE 526,&B00011000
90 VPOKE 527,&B00011000
100 PRINT"A"
```

U ziet nu dus een boompje in plaats van de letter 'A'!!

We gebruikten binaire getallen, omdat dit een beter overzicht geeft van de vorm van het karakter.

Als u de oude vorm terug wilt krijgen, kunt u dit doen

d.m.v. het terugpoken van de data voor de letter 'A', maar een eenvoudige screen n ($n=0, 1, 2, 3$) opdracht doet hetzelfde.

Bij scherm mode 0 vallen de rechter 2 bits van het karakter weg, in screen mode 1 is dit niet het geval.

HET GEBRUIK VAN VPOKE BIJ HET SCHERM

Het commando VPOKE kunt u gebruiken om een bepaalde locatie op het scherm te 'vullen' met de vorm van een bepaalde karaktercode. U kunt zelfs de anders niet beschikbare speciale karakters, zoals de 'smile' op deze manier op het scherm plaatsen.

Programma nr.7 laat de te VPOKEN karakters zien met de bijbehorende codes.

```
10 SCREEN0:KEY OFF:LOCATE,,0
20 FORX=0TO255
30 LOCATE11,12
40 PRINTX;"-":VPOKE500,X
50 FORY=1TO300:NEXTY
60 NEXTX
70 CLS:KEYON:LOCATE,,1
```

DE FUNKTIETOETSEN

Aan de onderkant van de tekstschermen staat de inhoud van de 10 functietoetsen van uw MSX computer.

KEY OFF schakelt deze display uit.

KEY ON schakelt deze display weer aan.

DEFUSR=&HOCC:X=USR(0) zet de display uit.

DEFUSR=&HOCCF:X=USR(0) zet de display aan.

Wilt u de inhoud van deze functietoetsen veranderen, dan doet u dit d.m.v. de instructie: **KEY N,"INHOUD"** waarbij n de waarde 1 tot 10 kan aannemen, overeenkomstig het nr. van de functietoets die de functie moet bevatten.

Voorbeeld:

KEY 1,"WIDTH 40"+CHR\$(13)

plaatst de opdracht **WIDTH 40** in functietoets nr.1 en drukt voor u meteen op de **RETURN/ENTER** toets (**CHR\$(13)** komt overeen met **ENTER/RETURN**).

CHR\$(13) is een van de vele control codes van uw MSX computer. Al deze codes kunt u gebruiken bij het definiëren van de functietoetsen. Een lijst van deze codes vindt u hieronder:

MSX CONTROL CODES

CHR\$ (2)	ctrl-B	- Beweeg de cursor naar het begin van het vorige woord.
CHR\$ (3)	ctrl-C	- Break en stop het programma.
CHR\$ (5)	ctrl-E	- Wis alles achter de cursor.
CHR\$ (6)	ctrl-F	- Beweeg de cursor naar het begin van het volgende woord.
CHR\$ (7)	ctrl-G	- Geeft een biep.
CHR\$ (8)	ctrl-H	- Zelfde als Backspace.
CHR\$ (9)	ctrl-I	- 8 spaties naar rechts.
CHR\$ (11)	ctrl-K	- Cursor naar de positie links bovenin het tekstscherf.
CHR\$ (12)	ctrl-L	- Wis het tekstscherf en zet de cursor in de positie links bovenin.
CHR\$ (13)	ctrl-M	- Vergelijk ENTER/RETURN.
CHR\$ (14)	ctrl-N	- Beweeg de cursor naar het eind van de regel.
CHR\$ (18)	ctrl-R	- Activeert de insert mode.
CHR\$ (21)	ctrl-U	- Wist de regel waarin de cursor zich bevindt.
CHR\$ (28)	ctrl-\	- Cursor naar rechts.
CHR\$ (29)	ctrl-]	- Cursor naar links.
CHR\$ (30)	ctrl-(SHIFT) (F6)	- Cursor omhoog.
CHR\$ (31)	ctrl-.	- Cursor naar beneden.

Om de functietoetsen in hun oorspronkelijke functie terug te krijgen gebruikt u:

```
DEFUSR=&H003E:X=USR(0) (ENTER)
```

Met VPOKE kunt u ook over de functietoets display heen schrijven. Programma nr.8 demonstreert deze mogelijkheid.

Programma nr.8:

```
10 KEYON:SCREEN0
20 FORX=920TO960STEP2
30 VPOKEX,0
40 NEXTX
50 END
```

(VIDEO-)GEHEUGEN INDELING

Het tekstschermb, screen 0, gebruikt het eerste gedeelte van het Video RAM om informatie op het scherm te plaatsen. Elke achtereenvolgende locatie van 0 t/m 959 representeert een karakterpositie op het scherm.

Er zijn echter in het gewone RAM geheugen ook enkele adressen die een nog grotere controle over het tekstschermb geven.

Eenvande beste instructies is misschien wel POKE&HF3B0,X (X=0 t/m 255). Hierdoor is het mogelijk om het aantal karakters dat op het tekstschermb moet verschijnen in te stellen, zonder centreren, zoals bij het WIDTH commando.

Een leuk grapje met dit commando:

- SCREEN 0
- POKE &HF3B0,255
- Zet de cursor voor een stukje tekst.
- Druk op insert.
- Druk op de spatiebalk.

De computer hangt zich nu echter op, gebruik nu uw reset-knop of indien uw computer er geen heeft, de aan/uitschakelaar.

Andere nuttige RAM adressen zijn:

ADRES	FUNCTIE
F3AE	Breedte van de regel in screen mode 0
F3AF	Breedte van de regel in screen mode 1

F3B0	Regellengte	
F3B2	Aantal regels op het scherm	
	SCREEN MODE 0	
F3B3	Start van de tabel van de screen locaties (name table) (0000H)	
F3B7	Start van de karakter tabel (Char. generator table) (0800H)	
	SCREEN MODE 1	
F3BD	Start van de tabel van de scherm locaties	- 1800H
F3BF	Start van de kleuren tabel	- 2000H
F3C1	Start van de karakter tabel	- 0000H
F3C3	Start van de attribute tabel	- 1B00H
F3C5	Start van de Sprite vorm tabel	- 3800H
	SCREEN MODE 2	
F3C7	Start van de tabel voor de scherm locaties	1800H
F3C9	Start van de kleuren tabel	- 2000H
F3CB	Start van de karakter tabel	- 0000H
F3CD	Start van de Attribute tabel	- 1B00H
F3CF	Start van de Sprite vorm tabel	- 3800H
	SCREEN MODE 3	
F3D1	Start van de tabel voor de schermlocaties	-0800H

F3D5	Start van de kleuren tabel	- 0000H
F3D7	Start van de attribute tabel	-1B00H
F3D9	Start van de Sprite vorm tabel	- 3800H

F3DB	Toets klik aan/uit (0=uit, groter of gelijk 1=aan)	
F3DC	Y-coördinaat van de cursor	
F3DD	X-coördinaat van de cursor	
F3DE	Functietoets display aan/uit (verg. F3DBH)	
F3DF t/m F3E6	Bevatten de waarden van de VDP registers 0 t/m 7	
F3E7	Status register van de VDP	
F3E9	Voorground kleur	
F3EA	Achtergrond kleur	
F3EB	Border kleur	
F55E	Input buffer	
F672	Hoogste geheugenlocatie	

DE VIDEO CHIP IN UW MSX COMPUTER

De MSX-1-reeks van computers maakt gebruik van een aparte chip (TMS 9918/9929 Video Display Processor, 'VDP') voor de opbouw van het scherm. Enkele bijzonderheden van deze video chip zijn de 32 sprites, pixelscroll en het aparte 16K video geheugen. De VDP gebruikt een 10738 Mhz kristal. De klokfrequentie van de Z80A wordt verkregen door 10738 te delen door 3.

Zoals u reeds weet, kunnen we het extra Video RAM van de VDP adresseren d.m.v. VPEEK en VPOKE.

De display modes zijn:

- 0 - tekstmode (2 kleuren max.)
- 1 - tekstmode (sprites mogelijk)
- 2 - high resolution mode (256*192)
- 3 - Multi Color mode (64*48)

In tekstmode 0 kunt u gebruik maken van maximaal 40 karakters op een regel.

In alle modes behalve mode 0 heeft u de beschikking over maximaal 32 karakters op een regel

Het Video RAM van 16384 bytes bevat vrijwel alle informatie die de computer nodig heeft om met graphics en tekst te kunnen manipuleren. Het bevat tabellen met informatie over verschillende screenmodes, tekstkarakters, spritevorm, en attribute-tabellen.

Hieronder vindt u de adressen in Video RAM met hun functie:

HET TEKSTSCHERM (SCREEN0)

Decimaal	Hexadecimaal	Beschrijving
0000-0959	0000H-03BFH	Karakter posities op 't scherm
2048-4096	0800H-1000H	Karakter vorm tabel

HET TEKSTSCHERM (SCREEN1)

Decimaal	Hexadecimaal	Beschrijving
0000-2047	0000H-07FFH	Karakter vorm tabel
6144-6911	1800H-1AFFH	Karakter posities op 't scherm
6912-7039	1B00H-1B7FH	Sprite attribute tabel
8192-8223	2000H-201FH	Kleuren tabel
14336-16383	3800H-3FFH	Sprite vorm tabel

HIGH RESOLUTION SCHERM (SCREEN2)

Decimaal	Hexadecimaal	Beschrijving
0000-6144	0000H-1800H	Karakter vorm tabel
6144-6911	1800H-1AFFH	Karakter posities op 't scherm
6912-7039	1B00H-1B7FH	Sprite attribute tabel
8192-14335	2000H-37FFH	Kleuren tabel
14336-15360	3800H-3C00H	Sprite vorm tabel

MULTI COLOUR SCHERM (SCREEN3)

Decimaal	Hexadecimaal	Beschrijving
0000-1535	0000H-05FFH	Karakter vorm tabel
2048-2816	0800H-0B00H	Karakter posities op 't scherm
6912-7039	1B00H-1B7FH	Sprite attribute tabel
14336-15360	3800H-3C00H	Sprite vorm tabel

Alle bovengenoemde adressen kunt u adresseren d.m.v. VPEEK en VPOKE.

Van deze locaties zullen we nu enkele wat nader toelichten, zodat u een beter inzicht krijgt, wat er nu allemaal mee gedaan kan worden.

UITLEG VAN DE SCREEN LOCATIES

SCREEN MODE 0

Bij het initialiseren van screen mode 0 worden de 256 karakters vanuit het ROM naar het Video RAM overgebracht. Ook wordt de regelbreedte ingesteld op 37 (F3AEH) en een CLS uitgevoerd en de kleuren ingesteld als color 15,4,4.

De karakter positie tabel voor screen mode 0 is 960 bytes lang. Het scherm wordt ingedeeld in hokjes. Het meest linkse hokje bovenaan het scherm heeft de waarde 0, eentje naar rechts heeft de waarde 1, etc. Elke regel heeft 40 karakters (in totaal 24 regels), dus het meest rechtse hokje van regel 1 heeft de waarde 39. Het hokje rechts onderin het scherm heeft dus logischerwijs de waarde 960 (24*40). Met de VPOKE instructie kunt u nu elke screen locatie vullen met het door u gewenste karakter.

Probleem: We willen in het midden van de eerste regel een smile plaatsen.

Opdracht: **VPOKE 19,1**

Resultaat: Plaatst de befaamde smile op locatie 19 van scherm mode 0.

Uitleg: De waarde 19 bevindt zich in de eerste regel van het scherm. Hierin plaatsen we de waarde van het daar te projékteren karakter. In dit geval wilden we een smile en aangezien deze als karakterwaarde een 1 heeft gekregen, komt de smile op het scherm.

De karakter vorm tabel is de tabel waarin de vorm van de karakters ligt opgeslagen. Een goed voorbeeld van de mogelijkheden van deze tabel wordt gegeven in het hoofdstuk „HET VERANDEREN VAN DE KARAKTERSET”.

SCREEN MODE 1

Deze tekstmode heeft in tegenstelling tot screen mode 0 slechts 32 karakters op een regel. Echter, het is ook nu mogelijk om sprites te gebruiken. Ook is met enkele trucjes de kleurenmogelijkheid veel groter.

Laten we eens kijken wat er zoal mogelijk is.

De 256 MSX karakters worden vanaf 1BBFH in ROM naar het Video RAM overgebracht. Het veranderen van de karakterset is gelijk aan de wijze zoals we dat al voor screen 0 hadden gezien. Ook het op het scherm plaatsen van de MSX karakterset via het VPOKE statement is gelijk aan scherm mode 0. Het enige verschil zijn de adressen, maar die staan in de tabel. Onthoud echter dat het scherm een maximale breedte heeft van 32 i.p.v. 40 in screen mode 0.

U kunt de kleuren van 8 opeenvolgende karakters aan uw wensen aanpassen d.m.v. het VPOKE statement. Stel dat u de letter 'A' een bepaalde kleur wilt geven.

De ASCII-code van de letter 'A' is gelijk aan 65. De integer-waarde van 65 gedeeld door 8 is gelijk aan 8. Dus het adres dat we moeten 'VPOKEN' komt overeen met de waarde $8192+8=8200$.

VPOKE 8200,&H3F

Levert ons een letter 'A' in het groen met een witte achtergrond. Zie voor een verklaring voor de hexadecimale notatiewijze appendix A.

Elke 'A' op het scherm zal deze kleuren voortaan aanne-

men. Echter ook de B, C, D, E, F en de G, omdat deze naar dezelfde plaats in de kleurentabel refereren.

Voor de verklaring van de Sprite attribute tabel en de Sprite vorm tabel verwijst ik naar het stuk over het high resolution scherm (screen2).

SCREEN MODE 2

Dit scherm stelt u in staat om met de vele grafische commando's die het MSX BASIC kent schitterende kleuren en tekeningen op het scherm te plaatsen.

Tekst is niet zomaar in 40 kolommen mogelijk, maar een slimme routine (zonder peeks en pokes) stelt u in staat dit wel te bewerkstelligen.

Deze routine neemt u als een subroutine op in uw programma's waardoor het veel prettiger wordt om tekst op screen 2 weer te geven. De routine volgt hieronder.

```
1 SCREEN2:OPEN"GRP:"AS1
10 X=?:Y=?:C=?:A$="UW TEKST":
   GOSUB100
15 GOTO 15
```

de routine:

```
100 COLOR C:FOR I=1 TO LEN(A$):B$=MID$(A$,
    I,1):X=X+6:PRESET(X,Y):PRINT#1,B$:NEXT
200 RETURN
```

In regel 10 geeft u 'x' en 'y' de waarde van de door u gekozen schermcoördinaten, waar de tekst moet komen. 'c' geeft u een waarde tussen de 0 en 15 voor de kleur van de te printen tekst. In 'a\$' stopt u de tekst en klaar is Kees!

De karakter vorm tabel is in deze mode 3 maal zo lang als in screen mode 1, n.l. 6144 bytes. Het scherm is echter ver-

deeld in 3 onafhankelijke delen.

Het eerste deel van het scherm komt overeen met de eerste 256 bytes van de karakter positie tabel. Elke waarde van een karakter in dit gedeelte van de karakter positie tabel wordt gedefinieerd door de eerste 2048 bytes van de karakter vorm tabel.

Het middelste deel van het scherm komt overeen met de 256ste t/m de 511e byte van de karakter positie tabel.

Bytes 512 t/m 767 komen overeen met het 3e deel van het scherm.

Dit alles komt er op neer, dat als je een bepaalde positie op het scherm een waarde geeft tussen de 0 en de 255, dan zal op die plaats het karakter verschijnen dat op die karakterpositie van het betreffende schermdeel staat.

Voorbeeld: VPOKE 6144,1:VPOKE 6146,1

Wanneer u nu een bepaald karakter op positie 8,0 van het scherm plaatst, dan zal dit karakter ook op de plaats 0,0 en 16,0 verschijnen, omdat deze ook de waarde 1 hebben (geVPOKED).

Op deze manier kunt u in machinetaalsnelheid een voorstelling op het scherm toveren. Een voorbeeld hiervan vindt u terug in programma nr.9:

```
10 'een menu kader voor MSX (VPOKE VERSI  
ON)  
20 :  
30 SCREEN2  
40 FOR X=6144 TO 6144+31  
50 VPOKEX,1  
60 NEXTX  
70 FOR Y=6144 TO 6144+(32*6)STEP32  
80 VPOKEY,1  
90 NEXT
```

```
100 FORY=6144+31TO6176+(31*6)STEP32
110 VPOKEY,1
120 NEXTY
130 FORX=6144+(32*6)TO((6144+(32*6))+31)
140 VPOKEX,1
150 NEXTX
160 OPEN"grp:"AS1
170 PRESET(8,0):PRINT#1,"*"
180 GOTO 180
```

SPRITES IN MODE 1, 2 EN 3

Sprites zijn een zeer belangrijke eigenschap van uw MSX computer.

Met deze zelf te definiëren figuurtjes kunt u heel mooie effecten bereiken, vooral in spelletjes.

Om nu wat meer controle over de sprites te krijgen, volgen hier voor alle 32 mogelijke sprites de adressen en hun functies.

SPRITE ATTRIBUTE TABEL

SPRITE NR.	Y-POSITIE	X-POSITIE	SPRITE PLANE	KLEUR
sprite 0	6912 (1B00H)	6913 (1B01H)	6914 (1B02H)	6915 (1B03H)
sprite 1	6916 (1B04H)	6917 (1B05H)	6918 (1B06H)	6919 (1B07H)
sprite 2	6920 (1B08H)	6921 (1B09H)	6922 (1B0AH)	6923 (1B0BH)
sprite 3	6924 (1B0CH)	6925 (1B0DH)	6926 (1B0EH)	6927 (1B0FH)
sprite 4	6928 (1B10H)	6929 (1B11H)	6930 (1B12H)	6631 (1B13H)
sprite 5	6932 (1B14H)	6933 (1B15H)	6934 (1B16H)	6935 (1B17H)
sprite 6	6936 (1B18H)	6937 (1B19H)	6938 (1B1AH)	6939 (1B1BH)
sprite 7	6940 (1B1CH)	6941 (1B1DH)	6942 (1B1EH)	6943 (1B1FH)
sprite 8	6944 (1B20H)	6945 (1B21H)	6946 (1B22H)	6947 (1B23H)
sprite 9	6948 (1B24H)	6949 (1B25H)	6950 (1B26H)	6951 (1B27H)
sprite 10	6952 (1B28H)	6953 (1B29H)	6954 (1B2AH)	6955 (1B2BH)
sprite 11	6956 (1B2CH)			
sprite 31	7036 (1B7CH)	7037 (1B7DH)	7038 (1B7EH)	7039 (1B7FH)

U kunt deze tabel gebruiken als naslag tijdens het programmeren.

Het volgende programma demonstreert het gebruik van de Sprite attribute tabel. Het maakt gebruik van de geheugenlocaties voor sprite 0 (zie de tabel).

Programma nr.10:

```
10 SCREEN2
20 FORX=1TO8
30 READA
40 A$=A$+CHR$(A)
```

```

50 NEXT
60 SPRITE$(0)=A$
70 PUTSPRITE0,(50,50),15,0
80 FORX=0TO255
90 VPOKE6913,X
100 VPOKE6915,RND(1)*12+2
110 NEXT
120 GOTO80
130 DATA 224,248,62,255,62,248,224,0

```

SPRITE VORM TABELLEN

De sprite vorm tabellen zijn afhankelijk van de door u gekozen sprite grootte.

SCREEN 2,0 - 8* 8 sprites

SCREEN 2,2 - 16* 8 sprites
16*16 sprites

8*8 sprite vorm tabel:

SPRITE NR.	DECIMAAL ADRES	HEX. DECIMAAL ADRES
sprite 0	14336 - 14343	3800H - 3807H
sprite 1	14344 - 14351	3808H - 380FH
sprite 2	14352 - 14359	3810H - 3817H
sprite 3	14360 - 14367	3818H - 381FH
sprite 4	14368 - 14375	 -
sprite 31	14584 - 14591	38F8H - 38FFH

16*8 sprite vorm tabel

SPRITE NR.	DECIMAAL ADRES	HEX. DECIMAAL ADRES
sprite 0	14336 - 14351	3800H - 380FH
sprite 1	14368 - 14383	3820H - 382FH
sprite 2	14400 - 14415	3840H - 384FH
sprite 3	14432 - 14473	3860H - 386FH
sprite 4	14464 - 14479	 -
sprite 31	15328 - 15343	3BE0H - 3BEFH

16*16 sprite vorm tabel

SPRITE NR.	DECIMAAL ADRES	HEX. DECIMAAL ADRES
sprite 0	14436 - 14367	3800H - 381FH
sprite 1	14368 - 14399	3820H - 383FH
sprite 2	14400 - 14431	3840H - 385FH
sprite 3	14432 - 14463	3860H - 387FH
sprite 4	14464 - 14495	 -
sprite 31	15328 - 15359	3BE0H - 3BFFH

Het volgende programma demonstreert de mogelijkheden van de sprite-animatie in combinatie met het VPOKE statement.

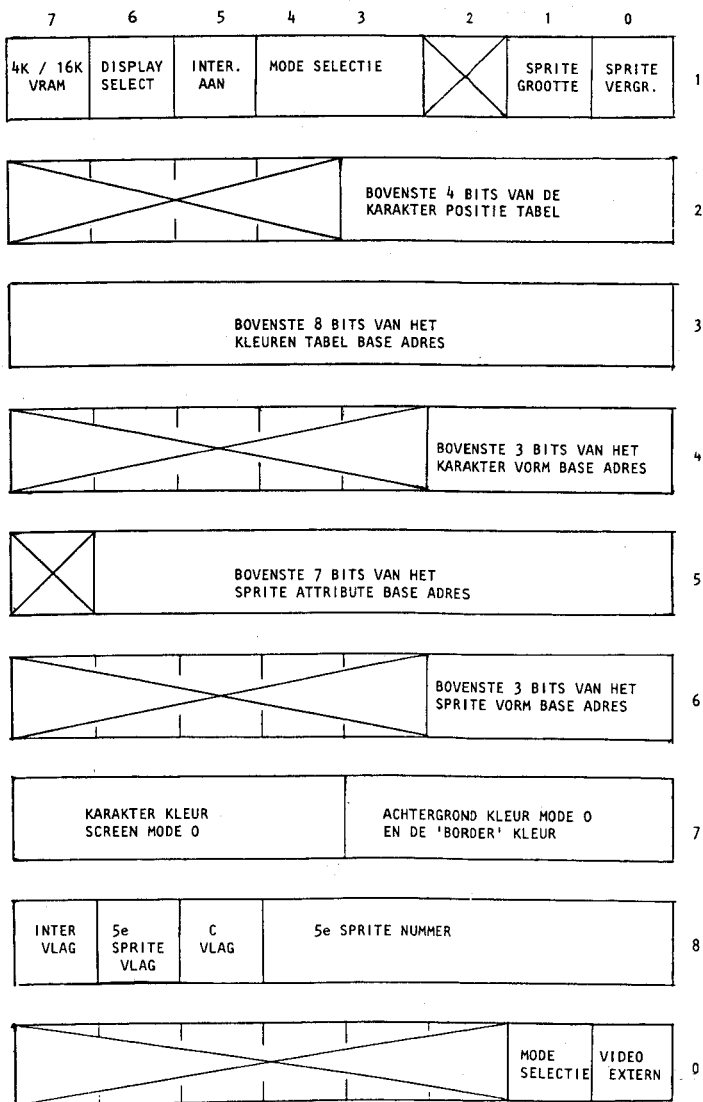
Programma nr.11:

```

10 COLOR15,1,1:SCREEN2,2
20 FORT=1T032:READA:A$=A$+CHR$(A):NEXT
30 SPRITE$(2)=A$
40 DATA 0,7,0,0,0,0,67,95,255,64,64,0,0,
0,0,0,0,255
50 DATA 32,112,200,196,226,226,242,124,0
,0,0,0,0,0
60 PUT SPRITE 2,(0,9),15,2
70 FORX=0T0255
80 VPOKE 6921,X
90 VPOKE &H3841,0:VPOKE&H3851,0:VPOKE&H3
846,3
100 VPOKE&H3847,31:VPOKE&H3849,0:VPOKE&H
384A,0
110 FORS=1T05:NEXTS
120 VPOKE&H3841,7:VPOKE&H3851,255:VPOKE&
H3846,67
130 VPOKE&H3847,95:VPOKE&H3849,64:VPOKE&
H384A,64
140 NEXT
150 GOT070

```

DE VDP REGISTERS



HET VDP STATEMENT

Zoals u reeds heeft kunnen lezen staat de term VDP voor Video Display Processor (In het Nederlands: Beeld verzorger). Deze VDP heeft om te kunnen werken, naast het Video RAM een aantal registers waar ook in binaire vorm informatie over uw MSX ligt opgeslagen. Voor binaire getallen zie appendix B.

Er bestaan in de TI9918/9929 9 registers. Deze registers zijn genummerd van 0 t/m 8. Al deze registers zijn geschikt om uit te lezen, of in te schrijven (met uitzondering van register nr.8). Al de registers zijn 8 bits groot (zie blz. 33).

Register nr.0

In dit register worden slechts de minst significante bits gebruikt (bits 0 en 1).

Bit 0: Deze bit staat normaal altijd op 0, tenzij er een externe VDP is aangesloten. U kent vast wel de mogelijkheid om d.m.v. een laserdisk interface een laserdisk speler op uw MSX aan te sluiten. Deze laserdisk speler functioneert dan als een externe VDP, dus zal deze bit op 1 staan. Probeer echter niet zelf iets met bit 0 uit te halen, want dan hangt uw computer zich op.

Bit 1: Deze bit wordt gebruikt in combinatie met bits 3 en 4 van VDP register nr.1 om de screen mode in te stellen.

Screen 0: (reg. 1, bit 4=1) (reg. 1, bit 3=0) (reg. 0, bit 1=0)

Screen 1: (reg. 1, bit 4=0) (reg. 1, bit 3=0) (reg. 0, bit 1=0)

Screen 2: (reg. 1, bit 4=0) (reg. 1, bit 3=0) (reg. 0, bit 1=1)

Screen 3: (reg. 1, bit 4=0) (reg. 1, bit 3=1) (reg. 0, bit 1=0)

Register nr.1

- Bit 0: Als dit bit op 0 staat dan zijn de sprites niet vergroot (unmagnified). Als dit bit op 1 staat dan zijn de sprites wel vergroot (magnified). Als u dit bit wilt gebruiken, dan moet u er rekening mee houden, dat het niet op zichzelf staat, maar in samenwerking met andere parameters functioneert. Het is namelijk afhankelijk van de screen mode selectie en de instelling van bit nr.1 van dit register.
- Bit 1: Als in het 'screen' commando wordt gekozen voor 8*8 sprites dan staat bit 1 op 0, als er wordt gekozen voor een 16*16 sprite, dan staat bit nr.1 op 1.
- Bit 2: Niet gebruikt.
- Bit 3: Zie bit 1 van register nr. 0.
- Bit 4: Zie bit 1 van register nr. 0.
- Bit 5: Dit is het VDP interrupt bit. Als dit bit op 0 staat voert de VDP geen interrupt meer uit. Aangezien dit interrupt signaal o.a. gebruikt wordt voor de keyboard scan, zal als u dit bit op 0 zet niets meer via het toetsenbord kunnen worden ingebracht. Een RESET of het uit- en aan schakelen van de computer is dan nog als enige oplossing over.
- Bit 6: Dit bit zorgt voor het aan- en uitschakelen van het beeldscherm. Als u nu een BASIC programmaatje schrijft, dat een cirkel tekent en in laat kleuren, dan kunt u voordat deze cirkel wordt getekent door bit 6 op 0 te zetten het scherm uit zetten.

Door na het paint commando het scherm weer aan te zetten (bit 6 op 1), komt er in 1 klap een cirkel op 't scherm.

Voorbeeld:

```
10 COLOR15,1,1:SCREEN2
20 VDP(1)=&B10100010
30 CIRCLE(128,96),60,15
40 PAINT(128,96),15
50 VDP(1)=&B11100010
60 GOTO 60
```

Bit 7: Hiermee kunt u kiezen tussen een 4K of 16K Video RAM:

0 - 4K

1 - 16K

Dit bit moet altijd op 1 staan.

Register nr.2

Alleen de 4 minst significante bits worden hier gebruikt. Dit 'nibble' wordt gebruikt om het 'BASE adres' van de karakter positie tabel te kunnen definiëren.

Het BASE adres is het eerste byte van de karakter positie tabel.

Het Video RAM is 16K groot, dus binair max. 14 bits groot. De nibble van register nr.2 definieert nu de meest significante nibble van een 14 bits getal.

Het karakter positie tabel BASE adres wordt nu gevormd op de volgende wijze: 00....00 00000000

Op de plaats van de puntjes komt de inhoud van het minst significante nibble van register nr.2.

3C00H - Max. BASE adres

0400H - Min. BASE adres

Screen m. 0: Karakt. pos. BASE adres=0000H ⇒ reg. nr.2=0
Screen m. 1: Karakt. pos. BASE adres=1800H ⇒ reg. nr.2=6
Screen m. 2: Karakt. pos. BASE adres=1800H ⇒ reg. nr.2=6
Screen m. 3: Karakt. pos. BASE adres=0800H ⇒ reg. nr.2=2

Register nr.3

Dit register bevat de bovenste 8 bits van het kleurentabel-base-adres. Deze tabel bevat de gegevens over de kleur van het te printen karakter. Niet over de achtergrond kleur of de border kleur. Deze tabel bestaat alleen in screen modes 1 en 2.

Het adres wordt als volgt opgebouwd:

00.....000000

Het max. base adres is dus 3FC0H

Het minimum is 0000H.

U kunt in stappen van 40H een eigen kleuren tabel in VRAM opzetten. (00000000 01000000 =40H)

Register nr.4

De minst significante 3 bits bepalen het karakter vorm BASE adres.

Het adres wordt als volgt opgebouwd:

00...000 00000000

Het maximum BASE adres is 3800H.

Het minimum BASE adres is 0000H.

De stap tussen de verschillende adressen is 0800H.

Register nr.5

De eerste 7 bits worden gebruikt om het Sprite attribute BASE adres te vormen.

Het adres wordt als volgt opgebouwd:

00..... 00000000

De maximum waarde is 3F80H.

De minimum waarde is 0000H.

De stap tussen de verschillende adressen is 0080H.

Register nr.6

De laagste 3 bits bepalen het sprite vorm BASE adres van de sprite vorm tabel.

Het adres wordt als volgt opgebouwd:

00...000 00000000

De maximum waarde is 3800H.

De minimum waarde is 0000H.

De stap tussen de verschillende adressen is 0800H.

Register nr.7

Het laagste nibble bepaalt de border kleur. In screen mode 0, ook de achtergrond kleur. De hoogste nibble bepaalt de voorgrond kleur.

Voorbeeld: $VDF(7) = \&HF1$

Geeft witte letters op een zwarte achtergrond.

F=15 (wit), 1=1 (zwart)
(vergelijk color 15,1)

Register nr.8

- Bit 7: Dit register wordt ook wel het VDP status register genoemd. Het kan alleen gelezen worden. Het meest significante bit is de interrupt status vlag. Het staat op 1 na een screen scan. Het staat op 0 als het status register is gelezen.
- Bit 6: Als er meer dan 5 sprites op een horizontale lijn staan wordt dit bit op 1 gezet. Het wordt op 0 gezet als het status register wordt gelezen.
- Bit 5: Dit bit wordt 1, als 2 sprites botsen. (vergelijk 'on sprite gosub' statement).
- Bits 0 t/m 4: Als 5 sprites op 1 horizontale lijn staan wordt hierin het plane nummer van de vijfde sprite gezet.

HET BASE STATEMENT

Het BASE statement geeft aan welk adres als begin van een tabel in Video RAM wordt gehanteerd. Ook kunt u met het BASE statement nieuwe BASE adressen aangeven voor de ook door u te maken tabellen.

Hieronder volgen de BASE adressen zoals standaard in uw MSX.

SCREEN 0

BASE (0) - Base van de tabel in tekst mode

BASE (1) - Niets

BASE (2) - Base van de karakter vorm tabel in tekst mode

BASE (3) - Niets

BASE (4) - Niets

SCREEN 1

BASE (5) - Base van de karakter positie tabel in tekst mode

BASE (6) - Base van de kleuren tabel in tekst mode

BASE (7) - Base van de karakter vorm tabel in tekst mode

BASE (8) - Base van de sprite attribute tabel in tekst mode

BASE (9) - Base van de sprite vorm tabel in tekst mode

SCREEN 2

BASE (10) - Base v.d. karakter positie tabel in high-res. mode

BASE (11) - Base v.d. kleuren tabel in high-res. mode

BASE (12) - Base v.d. karakter vorm tabel in high-res. mode

BASE (13) - Base v.d. sprite attribute tabel in high-res. mode

BASE (14) - Base v.d. sprite vorm tabel in high-res. mode

SCREEN 3

BASE (15) - Base v.d. karakter positie tabel in high-res. mode

BASE (16) - Niets

BASE (17) - Base v.d. karakter vorm tabel in multi-col. mode

BASE (18) - Base v.d. sprite attribute tabel in multi-col. mode

BASE (19) - Base v.d. sprite vorm tabel in multi-col. mode

APPENDIX A

HET HEXADECIMALE GETALLEN SYSTEEM

Normalerwijs maken wij gebruik van het decimale getallen systeem. Het grondtal voor dit systeem is 10 (0, 1, 2, 3, 4, 5, 6, 7, 8, 9).

Bij het hexadecimale getallensysteem maken we gebruik van 16 als grondtal (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F).

In uw computer bevinden zich verschillende manieren om makkelijk met hexadecimale getallen te kunnen werken.

Als u een hexadecimaal getal in uw BASIC programma wilt verwerken, dan moet aan dit getal vooraf gaan: '&H'.

Voorbeeld: **&HFF** (255 decimaal)

Wilt u een hexadecimaal getal omzetten in het voor u misschien makkelijker te begrijpen decimale systeem, moet u intikken:

```
===>> PRINT &HFF ==>> 255
```

Wilt u een decimaal getal omzetten in een hexadecimaal getal, dan moet u dit doen d.m.v. de functie HEX\$.

Voorbeeld: **PRINT HEX\$(255) (&HFF)**

N.B. In dit boekje vind u alle hexadecimale getallen voorzien van een 'H' achter het getal. Dit is gedaan, om de machinetaalprogrammeur van dienst te zijn. In een assembler wordt namelijk altijd op deze wijze met hexadecimale getallen gewerkt.

Bij de bespreking van de kleurenmogelijkheden onder screen mode 1, gebruikten we de opdracht: `VPOKE 8200,&H3F`. Hier kunt u zien dat een hexadecimale notatie zijn nut kan hebben. Waarom?

Omdat wij hier een groene letter op een witte achtergrond wilden verkrijgen. De kleur groen is bij de MSX o.a. gedefinieerd als de waarde 3. De kleur wit is gedefinieerd als 15 (&HF). Door nu de gewenste kleuren hexadecimaal te noteren en dan achter elkaar te schrijven verkrijgen we ons gewenste resultaat.

APPENDIX B

BINAIRE GETALLEN

De computer werkt alleen met nulletjes en eentjes, zoals u waarschijnlijk al weet. Het aantal beschikbare cijfers bij de binaire getallen is 2 (0, 1).

Een binair getal is bijvoorbeeld: 1010 (10 decimaal).

Wat is nu het verband tussen binair en decimaal?

1010 binair

... 0 $\Rightarrow$ 0

1010 binair;

1010 binair

... 0 $\Rightarrow$ $0 \cdot 2$ tot de macht 0 = 0

.. 1. $\Rightarrow$ $1 \cdot 2$ tot de macht 1 = 2

. 0. . $\Rightarrow$ $0 \cdot 2$ tot de macht 2 = 0

1. . . $\Rightarrow$ $1 \cdot 2$ tot de macht 2 = 8 +

10 decimaal

Decimaal naar binair omzetten

Hierbij moet u kijken welke macht van 2 als grootste voorkomt in het decimale getal, dat moet worden omgezet naar een binair getal. Dit getal trekt u van het uitgangsgetal af. De uitkomst van die aftreksom bekijkt u dan weer opnieuw enzovoort.

Voorbeeld: Het getal 329

Het grootste getal is $256 = 2$ tot de macht 8
 $329 - 256 = 73$

Het grootste getal is $64 = 2$ tot de macht 6
 $73 - 64 = 9$

Het grootste getal is $8 = 2$ tot de macht 3
 $9 - 8 = 1$

In het getal 329 komen dus voor:

1 * 2 tot de macht 8
0 * 2 tot de macht 7
1 * 2 tot de macht 6
0 * 2 tot de macht 5
0 * 2 tot de macht 4
1 * 2 tot de macht 3
0 * 2 tot de macht 2
0 * 2 tot de macht 1
1 * 2 tot de macht 0

Het binaire getal is nu dus: 101001001

Ook voor binaire getallen bestaan onder BASIC speciale voorzieningen.

Een binair getal wordt genoteerd als: &B101001001

Omzetten van decimaal naar binair:

```
PRINT BIN$(329) ==>> 101001001
```

Omzetten van binair naar decimaal:

```
PRINT &B101001001 ==>> 329
```

Binaire getallen worden vooral veel gebruikt bij het programmeren op het allerlaagste niveau. Bijvoorbeeld het veranderen van de inhoud van de PSG of de VDP registers.

N.B. Een byte is opgebouwd uit 8 bits (nullen en enen). Deze byte is te verdelen in twee 'nibbles' (een halve byte).

Zo'n nibble kan een maximale waarde van 15 (FH) aannemen. Een binair getal wordt dan ook omgezet in een hexadecimaal getal door elk van de nibbles in hex. vorm te noteren en achter elkaar te schrijven.

Voorbeeld: 11111010 binair
bestaat uit twee nibbles, n.l.: 1111 en 1010
Het eerste nibble is gelijk aan 15 decimaal (FH).
Het tweede nibble is gelijk aan 10 decimaal (0AH).

Het getal &B11111010 is dus gelijk aan &HFA.

PROGRAMMA'S

DISK LOADER UTILITY

```
5 COLOR 1,14:SCREEN0:WIDTH38
10 :
20 :
30 ' "Geef een machinetaal programma
40 ' "de toevoeging .BIN
50 ' "Hierdoor worden uw machinetaal
60 ' "programma's probleemloos opgestart
   vanuit dit programma
70 ' "BENOEM VERDER:
80 ' "Basic programma's met .BAS
90 ' "MSX DOS files met .COM/ .SYS
100 ' "Als het programma een .COM of een
   .SYS uitslag tegenkomt zal het MSX DOS
operating system opgestart moeten worden.
110 ' "Wanneer U dan de 'A>' op het sche
rm krijgt kunt U de naam van het door U
te laden programma als nog intikken en o
p ENTER/RETURN drukken. Het programma wo
rdt op deze manier 'gerund.'
120 PRINT"MSX DISK BASIC VERSION 1.0
130 PRINT"LOADER PROGRAM
140 PRINT"COPYRIGHT (c) 1985 H. KLOPPER
150 PRINT:PRINT"Druk een toets....
160 A$=INPUT$(1)
170 CLS
180 DIM A$(500)
190 COLOR1,14:KEY OFF:SCREEN0:WIDTH38
200 FILES
210 LOCATE 1,20:PRINT"Zet de cursor op h
```

```

et eerste karakter"
220 LOCATE 1,21:PRINT"van de file die u
wilt laden."
230 LOCATE 1,22,0:PRINT"Druk op ESC voor
het laden..."
240 Y=CSRLIN:X=POS(0) 'x & y cordinaat
250 BEGIN$=INPUT$(1) ' wacht op ESC
260 IF BEGIN$=CHR$(30)THENX=X:Y=Y-1:GOSU
B 510 'cursor up
270 IF BEGIN$=CHR$(28)THEN X=X+1:Y=Y:GOS
UB 510 'cursor right
280 IF BEGIN$=CHR$(29)THEN X=X-1:Y=Y:GOS
UB 510 'cursor left
290 IF BEGIN$=CHR$(31)THEN X=X:Y=Y+1:GOS
UB 510 'cursor down
300 IF BEGIN$=CHR$(27)THEN310 ELSE250
310 A=CSRLIN 'y - cordinaat
320 B=POS(0) 'x - cordinaat
330 F=A*40 ' schermcode
340 FORX=1 TO 12
350 CODE=VPEEK((P+B)+X) 'inlezen ascii
waarden
360 A$(X)=CHR$(CODE) 'ascii waarden
omzetten
370 IF CODE<>0 THEN NEXT ELSE 380
380 PROGRAM$=A$(1)+A$(2)+A$(3)+A$(4)+A$(
5)+A$(6)+A$(7)+A$(8)+A$(9)+A$(10)+A$(11)
+A$(12)
390 FOR X=800 TO 919:VPOKE X,&HDB:NEXT
' onderste 4 regels
400 TEST$=MID$(PROGRAM$,10,3)
410 IF TEST$="COM" THEN LOCATE 5,17:PRIN
TPROGRAM$:LOCATE5,18:PRINT"==> Command f
ile(MSX DOS)":GOSUB 560 :RUN
420 IF TEST$="SYS" THEN LOCATE 5,17:PRIN
TPROGRAM$:LOCATE 5,18:PRINT"==> Systeem

```

```

file(MSX DOS)":GOSUB 560 :RUN
430 IF TEST$="BAT" THEN LOCATE 5,17:PRINT
PROGRAM$:LOCATE 5,18:PRINT"==> Autostar
t file(MSX DOS)": GOSUB 560: RUN
440 IF TEST$="" THEN LOCATE 5,18:PRINT"G
een extension":FORT=1TO200:NEXT
450 IF TEST$="MSG" THEN LOCATE5,17:PRINT
PROGRAM$:LOCATE 5,18:PRINT"==> Message f
ile": GOSUB 560 :RUN
460 IF TEST$="DAT" THEN LOCATE5,17:PRINT
PROGRAM$:LOCATE 5,18:PRINT"==> Data file
": GOSUB 560 :RUN
470 IF TEST$="OVR" THEN LOCATE5,17:PRINT
PROGRAM$:LOCATE 5,18:PRINT"==> Overlay f
ile": GOSUB 560 :RUN
480 IF TEST$="BIN" THEN LOCATE5,17:PRINT
PROGRAM$:LOCATE 5,18:PRINT"==> Machineta
al ": GOSUB 560:GOSUB610
490 ON ERROR GOTO 580
500 CLS:COLOR1,14:LOCATE5,5:PRINT"De fil
e "+PROGRAM$:PRINT" wordt nu geladen...
":LOADPROGRAM$:IF ERR=57THEN BLOADPROGRA
M$,R ' laden v/d file
510 IF Y>=19THENY=19 /\
520 IF X=-1 THEN X=0 '_Cursor niet
530 IF Y=-1 THEN Y=0 '/ uit scherm
540 LOCATEX,Y
550 RETURN
560 FOR T=1TO1000:NEXT
570 RETURN
580 LOCATE 5,16:PRINT"Fout seconstateerd
...."
590 FORT=1TO500:NEXT:RESUME 600
600 RUN
610 CLS:LOCATE5,5:PRINT"De file "+PROGRA
M$:PRINT"wordt nu geladen...":BLOADPROGR
AM$,R

```

ANIMATIE MET KARAKTERS

```
10 KEYOFF
20 COLOR 1,14,14:SCREEN1:WIDTH32
30 :
40 ' Y=X-COORDINAAT
50 ' X=Y-COORDINAAT
60 :Y=10
70 A%=0
80 B%=0
90 FOR I=1TO 8
100 READ A%
110 VPOKE (0+(250*8))+I,A%
120 NEXT
130 DATA 0,129,166,24,0,0,0,0
140 :
150 FORI=0TO7
160 READB%
170 VPOKE (251*8)+I,B%
180 NEXT
190 DATA 0,255,0,0,0,0,0,0
200 FOR X=20 TO5STEP-1:Y=Y+1
210 VPOKE 6144+(X*32)+Y,250
220 FORT=1TO100:NEXT
230 VPOKE 6144+(X*32)+Y,251
240 FORT=1TO100:NEXT
250 VPOKE 6144+(X*32)+Y,0
260 NEXT
```

LIJNEN VERDWIJNEN

```

10 STOP ON:ONSTOP GOSUB 200
50 X1=ABS(&H8000)
60 X2=PEEK(63170!)+256*PEEK(63171!)
70 FORY=X1TOX2
80 IFPEEK(Y)=143ANDPEEK(Y+1)=90THENC=1:N
EXT
90 IFC=1ANDPEEK(Y)=90THENPOKEY,127ELSEC=
0
100 NEXT
110 DELETE 50-110
120 A$="MSX":REMZZZZZZZZZZZZZZZZZZZZZZZZ
130 CLS:INPUT"GEEF PASSWORD":B$
140 IFB$<>A$THENPRINT"HELAAS...":NEW:REM
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
200 RETURN

```

BASIC GEHEUGEN

(uitleg van bovenstaand programma)

Het start adres van BASIC ligt altijd op &H8000 bij een 64K MSX.

De lengte van een BASIC programma is variabel, dus ook het eindadres. Het einde van een BASIC programma ligt bij het begin van de variabelen tabel. (X2)

U vindt het misschien nuttig, om lijnen in uw programma's te verbergen voor andermans nieuwsgierige ogen.

Bovenstaand programma zorgt voor dit fenomeen.
Typ het programma over en SAVE het op cassette of disk.
Alleen lijn nr. 130 blijft in de listing over. Als u nu het pro-

gramma 'RUNT', moet u het goede Password invoeren, anders verdwijnt het programma.

Het programma verwisselt de letters 'Z' met een delete teken.

BEELDSCHERM WEGSCHRIJVEN

```
1 COLOR 15,1,1:SCREEN2
2 FORX=1TO255STEP3
3 LINE(150,X)-(X,50),15
4 NEXT
40 TIME=0
50 :
60 BSAVE "pict1", O, &H3FFF, S      ' Be
eldscherm wegschrijven
70 :
80 SCREEN 0
90 PRINT "TIJD = ";TIME/50;"Seconden"
100 PRINT "Het beeld is nu weggeschreven"
110 :
120 INPUT"Wilt U het scherm weer laden (
j/n)";SCHERM$
130 IF SCHERM$="j"ORSCHERM$="Ja"OR SCHER
M$="J"THEN 150
140 IF SCHERM$="n"ORSCHERM$="nee"OR SCHE
RM$="N"THEN END
150 SCREEN 2:BLOAD "pict1", S      ' Beel
dscherm laden
160 :
170 BEEP
180 GOTO 180
```

HEADERLEZERDISK UTILITY****

De meeste mensen die in het bezit zijn van een diskdrive zullen zich afvragen, waarom de meeste software toch altijd op die langzame cassettes wordt uitgebracht.

Het antwoord is simpel. De meeste MSX gebruikers bezitten een cassetterecorder. Het aantal mensen dat zich de gelukkige bezitter van een diskdrive mag noemen is dus te klein voor de softwarehuizen om hun software op diskette uit te brengen.

Dit zal echter wel veranderen in de toekomst, want de meeste fabrikanten komen binnenkort met MSX machines met een ingebouwde diskdrive (vgl. de SVI X' Press).

Deze computers zijn dus in basisuitvoering al van een diskdrive voorzien.

Echter tot die tijd is het ploeteren met cassette bandjes terwijl u naast uw computer zo'n mooie snelle diskdrive hebt staan.

Wat zou het toch makkelijk zijn als u al uw cassette software op disk kwijt kon!

Met BASIC programma's (tot een KB of 20) geeft dit geen problemen. Echter met machinetaal programma's is dit moeilijker. Hiervoor volgt dadelijk een programma dat is geschreven in machinetaal/BASIC.

Het programma stelt u in staat om vrijwel elk machinetaal programma van cassette naar disk te schrijven.

Elk machinetaal programma heeft een BEGIN-, EIND- en START-adres.

Normaal laadt u een machinetaal programma met:

```
BLOAD"CAS:",R
```

Om nu een machinetaal programma van cassette naar disk te schrijven dient u eerst het programma op de vol-

gende bladzijde te 'RUNNEN'.

- U spoelt uw cassette met het te copieren programma terug en drukt op PLAY.
- Nu leest het programma de zogenaamde HEADER van de tape.
- Er verschijnen gegevens over het programma;
n.l.:
 - Soort programmeertaal (BASIC (Progr./ASCII of machinetaal))
 - Naam van het programma
 - Begin-, eind- en startadres

U kunt nu het programma wegschrijven d.m.v.:

```
BSAVE"PROGRAMMANAAM",&HBEGINADRES,  
&HEINDADRES,&HSTARTADRES
```

P.S. Programma's waarvan het begin-, eind-, of startadres hoger ligt dan DF87H kunnen problemen opleveren met de Diskbasic.

```
1 COLOR 1,14:SCREEN0:WIDTH40  
20 RESTORE  
30 READ A$ :CLEAR 300,VAL("&H"+A$):READ  
A$  
40 READ A$ : IF A$="**" THEN 80 ELSE I%=  
VAL("&H"+A$)  
50 READ A$: IF A$="**" THEN 40 ELSE POKE  
I%,VAL("&H"+A$) : I%=I%+1 : GOTO 50  
60 DATA cfff  
70 DATA d000,cd,e9,72,21,00,d1,06,10,cd,  
d4,72,77,23,10,f9,3e,02,32,63,f6,c9,e7,0  
,**,d040,21,9a,20,cd,e1,00,3e,2,32,93,f7  
,c9,**,**  
80 DEFUSR0=&HD000 : DEFUSR1=&HD040  
+ 90 DEFFNH$(V%)=STRING$(2-LEN(HEX$(V%)),"  
0")+HEX$(V%) 'RIGHT$("00"+HEX$(V%),2)  
54
```

```

100 PRINT "header  d3=prog,  ea=ascii,
d0=bin      filename " / " screenmodus"
110 PRINT STRING$(38,"-")
120 A=USR(0):  IF PEEK(&HD100)<128 THEN
120
125 FOR I=1 TO 500 : NEXT
130 FOR I%=0 TO 9 : PRINT FNH$(PEEK(&HD1
00+I%));" " : NEXT :PRINT
135 FOR I%=0 TO 5 : PRINT FNH$(PEEK(&HD1
0A+I%));" " : NEXT
140 PRINT "      | >";
150 FOR I%=0 TO 5 : A=PEEK(&HD10A+I%):IF
(A>31) AND (A<127) THEN PRINT CHR$(A);
: NEXT ELSE PRINT " " : NEXT
160 PRINT "'      |      ";FNH$(PEEK(&HD11
0))
170 L=PEEK(&HD100):A=USR(0) : IF L<>&HDO
THEN 120
175 FOR I=1 TO 500 : NEXT
180 PRINT "  Begin  ";FNH$(PEEK(&HD101))
;FNH$(PEEK(&HD100));
190 PRINT "   eind   ";FNH$(PEEK(&HD103))
;FNH$(PEEK(&HD102));
200 PRINT "   start  ";FNH$(PEEK(&HD105))
;FNH$(PEEK(&HD104))
210 GOTO 120

```

ROM-ROUTINES

De ROM-routines kunt u in sommige gevallen gebruiken in uw BASIC programma's en in elk geval in uw machinetaal-programma's.

In een BASIC programma doet u dat d.m.v. het DEFUSR en het USR statement.

Voorbeeld:

```
DEFUSR=&H000:XUSR(0) ==>> WARME START
```

In een machinetaalprogramma doen we dat d.m.v. een z.g. CALL-opdracht.

Hieronder volgen de ROM-routines en hun functies:

ADRES	UITLEG
0000H	RESET Als u deze routine aanspreekt zal de computer een warme start uitvoeren.
0008H	RST 08H Zelfde als boven, echter nu wordt gekeken of het in register HL staande byte gelijk is aan het byte dat op deze instructie volgt.
000CH	LD A, (HL) slotselectie Register A bevat het slotnummer. Als resultaat bevatten register A en register E de naar adres HL verwijzende slotwaarde.
0010H	VOLGENDE BYTE Zoekt volgende karakter of BASIC teken.

HL verwijst naar het volgende karakter.
A bevat het karakter.
Carry vlag gezet als het een getal is.
Zero vlag gezet als het einde is bereikt.

- 0014H LD (HL), E SLOTSELECTIE
A bevat het slotnummer, HL het adres en E de waarde.
- 0018H RST 18H
Print de waarde van register A op de 'listdevice' (normaal het beeldscherm).
Als u het adres F416 met een waarde groter dan 0 laadt, zal de printer worden ingeschakeld.
- 0020H RST 20H
DE en HL worden met elkaar vergeleken.
DE en HL worden niet veranderd.
- 0028H RST 28H
Geeft het gebruikte type variabele.
- 0038H RST 38H
Interrupt routine (50 keer per seconde)
- 003EH Functietoetsen nemen hun oude waarde aan.
- 0041H DISABLES SCREEN
Zet het scherm uit.
De schermkleur neemt de borderkleur aan.
- 0044H ENABLES SCREEN
Zet het scherm aan.
- 0047H Schrijf naar een VDP register
Register C bevat het registernummer.
Register B bevat de data.
In BASIC: VDP(C)=B

- 004AH Lees uit het Video RAM
 Het adres uit HL wordt in de Accumulator
 geladen.
 In BASIC: A=VPEEK(HL)
- 004DH Schrijf naar het Video RAM
 De waarde in de Accumulator wordt in HL
 van het Video RAM geladen.
 In BASIC: VPOKE HL,A
- 0050H VDP READ INITIALISATIE
 HL bevat het start adres.
- 0053H VDP WRITE INITIALISATIE
 HL bevat het begin adres.
- 0056H SCHRIJF NAAR VRAM
 Start in HL
 Data in A
 Lengte in BC
- 0059H VRAM naar RAM
 Verplaatst een blok VRAM naar RAM
 Startadres VRAM in HL
 Doeladres in RAM in DE
 Lengte in BC
- 005CH RAM naar VRAM
 Startadres RAM in HL
 Doeladres in VRAM en DE
 Lengte in BC
- 005FH Screen mode selectie
 De Accumulator bevat de waarde van de screen-
 mode.
 In BASIC: SCREEN A (met A=0, 1, 2 of 3).
- 0062H KLEUR INITIALISATIE
 Kleur als:

Voorggrondkleur (F3E9H)
Achtergrondkleur (F3EAH)
Borderkleur (F3EBH)
In BASIC: COLOR A,B,C

0066H	NON MASKABLE INTERRUPT
0069H	SPRITE INITIALISATIE
006CH	SCREEN 0 INITIALISATIE In BASIC: SCREEN 0
006FH	SCREEN 1 INITIALISATIE In BASIC: SCREEN 1
0072H	SCREEN 2 INITIALISATIE In BASIC: SCREEN 2
0075H	SCREEN 3 INITIALISATIE In BASIC: SCREEN 3
0078H	INITIALISATIE VDP SCREEN 0
007BH	INITIALISATIE VDP SCREEN 1
007EH	INITIALISATIE VDP SCREEN 2
0081H	INITIALISATIE VDP SCREEN 3
0084H	SPRITE VORM TABEL HL bevat het adres van de sprite vorm tabel In register A stopt u het sprite nummer.
0087H	SPRITE ATTRIBUTE TABEL HL bevat het adres van de sprite attribute tabel In register A stopt u het sprite nummer.
008AH	SPRITE GROOTTE. Register A bevat de sprite grootte. Carry-vlag gezet als 16x16 sprite.

008DH	Schrijft karakter naar graphics scherm Waarde van karakter in Accumulator.
0090H	Initialisatie PSG
0093H	Schrijf naar een PSG register Schrijft de waarde van register E in het PSG- register met de waarde 'A'. In BASIC: SOUND A,E
0096H	Lees uit een PSG register De Accumulator bevat de waarde van het PSG- register.
0099H	Start de achtergrondmuziek
009CH	KEYPRESSED Kijkt of er een toets wordt ingedrukt. Zero-vlag geset als toets in buffer.
009FH	Wacht op een toetsdruk Wacht tot er een toets wordt ingedrukt. De waarde van het karakter wordt in de Ac- cumulator gestopt. In BASIC: INKEY\$/INPUT\$(1)
00A2H	Print inhoud Accumulator op het scherm
00A5H	Print inhoud Accumulator op de printer
00A8H	PRINTER CONTROLE FFH in de Accumulator, en Zero-vlag gereset als printer 'ON-LINE'. Zero-vlag gezet als printer niet 'ON-LINE'.
00AEH	Invoer tot RETURN/ENTER U kunt een regel laten invoeren tot maximaal 255 karakters. In het bereik van F55EH tot F65DH.

	HL bevat het startadres -1.
00B7H	CTRL/STOP controle Carry-vlag geset als ingedrukt. In BASIC: STOP ON:ON STOP GOSUB
00C0H	BEEP Geeft een biep-geluidje In BASIC: BEEP
00C3H	CLS Wist het scherm en plaatst de cursor in de linker bovenhoek. In BASIC: CLS
006CH	LOCATE (CURSOR POSITIE) Register H - regel Register L - kolom In BASIC: LOCATE L,H
00C9H	FUNCTIE TOETSEN AAN OF UIT? Zet functietoetsen aan als ze aan moeten.
00CCH	FUNCTIE TOETSEN UIT Schakelt de functietoets-display uit. In BASIC: KEY OFF
00CFH	FUNCTIE TOETSEN AAN Schakelt de functietoets-display aan. In BASIC: KEY ON
00D2H	TEKST MODE Zet de computer in de tekst mode.
00D5H	OPVRAGEN VAN DE JOYSTICK STATUS Hierdoor komt in de Accumulator de waarde van de richting te staan waarin de joystick wordt bewogen. In BASIC: A=STICK(A)

00D8H	VUURKNOP ROUTINE In de Accumulator staat een waarde. Accumulator 0 - niets Accumulator 255 - toets ingedrukt
0132H	HOOFDLETTERS AAN/UIT Accumulator = 0 == > hoofdletters uit Accumulator < > 0 == > hoofletters aan
0156H	Wist KEYBOARD BUFFER
013EH	LEEST HET STATUS REGISTER De Accumulator bevat de waarde (0-255).

MSX-LOGO

```
10 ' MSX-LOGO
20 :
30 :
40 COLOR 3,5,5
45 :
50 'm
55 :
60 C=15:SCREEN2:LINE(5,160)-(25,0),C
70 LINE(25,0)-(45,0),C
80 LINE-(55,90),C
90 LINE-(70,0),C
100 LINE-(95,0),C
110 LINE-(110,130),C
120 LINE(110,160)-(85,160),C
130 LINE-(78,95),C
140 LINE-(70,160),C
150 LINE-(45,160),C
160 LINE-(38,95),C
170 LINE-(30,160),C
180 LINE-(5,160),C
190 :
200 's
210 :
220 CIRCLE(140,64),15,C,.5*3.14,1.5*3.14
    ,1.25
230 LINE(139,79)-(145,79),C
240 CIRCLE(145,120),40,C,1.5*3.14,.5*3.14,1.25
250 LINE(145,160)-(100,160),C
260 LINE(140,48)-(170,48),C
270 LINE(108,130)-(140,130),C
280 CIRCLE(140,122),8,C,1.5*3.14,.5*3.14
```

```

,1.25
290 LINE(140,114)-(130,114),C
300 CIRCLE(132,57),56,C,.5*3.14,1.5*3.14
,1.9/1
310 LINE(132,0)-(170,0),C
330 :
340 '   x
350 :
360 LINE(170,0)-(256,160),C
370 LINE(170,48)-(230,160),C
380 LINE-(256,160),C
390 LINE(230,0)-(170,160),C
400 LINE(256,0)-(197,160),C
410 LINE-(170,160),C
420 LINE(256,0)-(230,0),C
430 CIRCLE(200,100),2,5
440 CIRCLE(210,80),2,5
450 CIRCLE(200,75),2,5
460 PAINT(85,2),C
470 PAINT(235,140),C
480 PAINT(215,100),C
490 PAINT(190,140),C
500 GOTO 500

```

SCROLL

```
10 ' DIT PROGRAMMA LAAT ZIEN HOE U
20 ' EEN SCHERM KUNT LATEN VER-
30 ' PLAATSEN.
40 :
50 ' DIT ZOU EIGENLIJK IN MACHINE-
60 ' TAAL MOETEN GEBEUREN.
70 :
80 ' DIT MACHINETAALPROGRAMMA KUNT
90 ' U TERUGVINDEN IN HET
100 ' MACHINETAAL BOEK VAN
110 ' H. KLOPPER EN M. LEBELLE
120 ' STARK-TEXEL
130 :
140 DEFINT A-Z
150 SCREEN 2
160 LINE (0,0)-(255,191)
170 CIRCLE (200,96),50:PAINTSTEP(0,0)
180 :
190 FOR Y=&H1800 TO &H1AFF
200 Z=VPEEK(Y)
210 A=(Z+1) AND &H3F:Z=(Z AND &HE0) OR A
220 VPoke Y,Z
230 NEXT :GOTO 190
```

SIMULATIE OPSTARTSCHERM

```
10 ' DIT PROGRAMMA SIMULEERT ONDER
20 ' SCREEN MODE-1 HET OPSTART SCHERM
30 ' VAN UW MSX COMPUTER....
40 :
50 X=24:Y=39
60 STOPON:ONSTOPGOSUB300
70 WIDTH30
80 SCREEN1,0:VDP(6)=VDP(4)
90 VPOKE&H201F,0
100 PRINT" MSX BASIC version 1.0
110 PRINT" Copyright 1983 by Microsoft
120 LOCATE,2:PRINTFRE(0)" Bytes free"
130 :
140 ' haal de REM uit regel 150 als u ee
n diskdrive heeft
150 'LOCATE,3:PRINT" Disk BASIC version
1.0"
160 LOCATE,,0:PUTSPRITE0,(X,Y),15,255
170 A$=INPUT$(1)
180 VPOKE &H201F,0
190 IFA$=CHR$(&H1E) THEN X=X:Y=Y-8:GOTO
250
200 IFA$=CHR$(&H1F) THEN X=X:Y=Y+8:GOTO
250
210 IFA$=CHR$(&H1D) THEN X=X-8:Y=Y:GOTO
250
220 IFA$=CHR$(&H1C) THEN X=X+8:Y=Y:GOTO
250
230 IFA$=CHR$(&HC) THEN Y=7:X=16:CLS:GOTO
160:GOTO250
240 B=Y/8+1: LOCATEX/8-1,B:PRINTA$:: X=X
+8
```

```

250 IFY=>184THENY=183
260 IFY=<-2 THENY=-1
270 IFX=<16THENX=240:Y=Y-8:GOTO 290
280 IFX=>248THENX=16:Y=Y+8
290 GOTO 160
300 :
310 'Letters ALLE KLEUREN + Programma we
320 :
330 FORX=&H2000 TO&H201F
340 VPOKE X,1
350 NEXT
360 PUTSPRITE 0,(10,209),0,255
370 LOCATE 1,5:PRINT"Computer Internal e
rror
380 LOCATE 1,6:PRINT"Thank you!
390 LOCATE 1,8:PRINT"P.S. Druk op reset
of zet uw
400 LOCATE 1,9:PRINT"computer aan en uit
.....
410 STOP ON:ON STOP GOSUB440
420 FORT=1TO3000:NEXT
430 DEFUSR=0:X=USR(0)
440 RUN

```

WILLEKEURIGE KLEUREN

```
10 / DIT PROGRAMMA PLAATST
20 / WILLEKEURIG KLEUREN
30 / IN DE KLEUREN TABEL
40 / VAN SCHERMMODE 1...
50 :
60 SCREEN1
70 FORX=&H2000 TO&H201F
80 VPOKE X,RND(1)*12+2
90 NEXT
```

BALSPEL

```
10 ' EEN FRAAIE BASIS
20 ' VOOR EEN BALSPELPROGRAMMA
30 DEFINT A-Z
40 SCREEN 2,2
50 FOR S=0 TO 63
60 READ X
70 VPOKE BASE(14)+S,X
80 NEXT
90 FOR M=2 TO 4
100 FOR S=1 TO 16
110 READ X,Y
120 S$=S$+CHR$(X) : T$=T$+CHR$(Y)
130 NEXT
140 SPRITE$(M)=S$+T$
150 S$="" : T$=""
160 NEXT
170 S$=""
180 FOR S=1 TO 8
190 READ X
200 S$=S$+CHR$(X)
210 NEXT
220 SPRITE$(5)=S$
230 LINE (25,45)-(230, 45),15
240 LINE -(250,179),15
250 LINE -( 5,179),15
260 LINE -( 25, 45),15
270 LINE(127,45)-(127,179),15
280 CIRCLE (127,102),20,15
290 CIRCLE (127,102), 1,15
300 A=0:B=1:C=2:D=3:E=4:F=18:G=8
310 X=150:Y=100:Z=16:X1=100:Y1=100
320 PUT SPRITE C,(X,Y),15
```

```

330 PUT SPRITE D,(X,Y+Z),15
340 PUT SPRITE E,(X,Y),1
350 PUT SPRITE 5,(X1,Y1),15
360 ON STICK(0) GOSUB 470,470,480,480,49
0,490,500,500
370 ON STRIG GOSUB 610
380 IF X=X1 THEN GOSUB 580
390 IF X+1=X1 THEN GOSUB 580
400 IF X+2=X1 THEN GOSUB 580
410 IF X-2=X1 THEN GOSUB 580
420 IF X-1=X1 THEN GOSUB 580
430 SWAP F,G
440 PUT SPRITE 5,(X1,Y1+F),15
450 FOR T=1 TO 100 : NEXT
460 GOTO 360
470 Y=Y-2:GOSUB 510:RETURN
480 X=X+2:GOSUB 510:RETURN
490 Y=Y+2:GOSUB 510:RETURN
500 X=X-2:GOSUB 510:RETURN
510 PUTSPRITE C,(0,196),0
520 PUTSPRITE D,(20,196),0
530 SWAP A,C:SWAP B,D
540 PUTSPRITE C,(X,Y),15
550 PUTSPRITE D,(X,Y+Z),15
560 PUTSPRITE E,(X,Y),1
570 RETURN
580 X1=X:Y1=Y
590 STRIG(0) ON
600 RETURN
610 FOR T=1 TO 20
620 X1=X1-1:Y1=Y1-1
630 PUTSPRITE 5,(X1,Y1),15
640 NEXT
650 FOR T=1 TO 20
660 X1=X1-1:Y1=Y1+1

```

```

670 PUTSPRITE 5,(X1,Y1),15
680 NEXT
690 STRIG(0) OFF
700 RETURN
710 DATA 0, 0, 0, 0, 0, 0, 0, 0, 3,
      0, 0, 0, 0, 0, 0, 0, 0
720 DATA 0, 20, 60, 20, 12, 18, 45, 173,
249, 113, 33, 33, 30, 30, 30, 62
730 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0
740 DATA 54, 102, 70, 102, 50, 18, 51, 7,
      0, 0, 0, 0, 0, 0, 0, 0
750 DATA 0, 0, 0, 20, 0, 60, 0, 20,
      0, 12, 0, 18, 1, 45, 1, 173
760 DATA 0, 249, 0, 113, 0, 33, 0, 33,
      0, 30, 0, 30, 0, 14, 0, 14
770 DATA 0, 14, 0, 14, 0, 12, 0, 12,
      0, 12, 0, 12, 0, 28, 0, 60
780 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0
790 DATA 0, 14, 0, 3, 0, 3, 0, 2,
      0, 0, 0, 12, 0, 18, 0, 18
800 DATA 0, 6, 0, 14, 0, 30, 0, 30,
      0, 0, 0, 0, 0, 0, 0, 0
810 DATA &B00000000
820 DATA &B00000000
830 DATA &B00000110
840 DATA &B00001111
850 DATA &B00001111
860 DATA &B00000110
870 DATA &B00000000
880 DATA &B00000000

```

WERELDKAART

```
10 / DIT PROGRAMMA TEKENT DE
20 / KAART VAN DE WERELD
30 :
40 / AAN U DE TAAK OM ER IETS
50 / NUTTIGS MEE TE DOEN....
60 /
70 :
80 COLOR15,4,4
90 SCREEN 2
100 CIRCLE(124,100),155,5,,,19/31
110 PAINT(5,5),5:PAINT(250,180),5:PAINT(
5,180),5
120 CLEAR1000
130 A$="C15BM8,43R1BR1R1U1R3U1R3U2L2U1R2
U2R2D1R3U2L1U1L1R1U1R2D1R3U2L1R4U1R4D1R2
D1R3D2R2U1R5D2L3D2R1D3L1D2R1D2R3D2R3U2L2
U2L3R3U2L2U1L1R1U1R2U3L1U1R1U1D1R1D1R1D1
R3D1G1L2D2R2U2E1R2U1R2D1R3U1D2"
140 B$="R2U1R1U2R1U1R1U1R1U1R1L1D1L1D2L1
D3R4U1L1U1D1L1D1L3D1F2H1G1E1H1U1L2D2L3D1
L2D1L2D1L1D5R1D1R2D1R1D4R2U2R1U2R2U1H1F1
R1U3R1U1R1U1H1F1R1U1D1R3D4R2U2R1D5R1D1R1
D1L2D1L7D1L2D2R4L1D3R1L2G1E1U1L3D1L2D1L1
D1L1U3"
150 C$="L3U2L2U2L1U1L2D1L3D3R2U1R2L2D1L1
D1L1D4R2U1R1U2R1U1D2R1L2D2R4U3L1U1D1R2D1
R4D2L1D1L1D1L2D2L1D2L2D1L2D1L2D1L1D6L1D1
U6L5D1L6D1L1D1L1D2R1L1D1L1D5R1D1R5U1R1U1
R2L1D2L1D3R3D4L3U2L3U1L1U1L1U1"
160 D$="L4U1L1U1L2U1L1U1L1U1R1U2L1U4L1U6
L2D9U4L1R1U2L1U6L1U5R1U2R1U2R2U2R1U1R1U2
R2U5R1U4R1U2L1U1L1U2L2U1L2D1L1D1L1R1U2L3
```

D1L1D1L2"
 170 DRAW A\$
 180 DRAW B\$
 190 DRAW C\$
 200 DRAW D\$
 210 A\$="":B\$="":C\$="":D\$=""
 220 A\$="BM41,94L1D1R2D1L1R2D1R1U1R2D2R1D
 3L1D3L1D1L1D3R1D2L1D2R1D1R1D2R1D2R1D
 2R2D1R1D1R1D1R1D2R1D9R1D8R1D3R1D1R1D5L1R
 1D1R1D1R1D2R1D1R1D1R1D1U2R1F1D2R3L1U1L1H
 2L1U2R1U3L2U2R1U3"
 230 B\$="L1U2R2U2R2U1R2U1D1L2U1L1U2R4U1R1
 U3R1U5R1U1R1U1R3U1R1U3R1U1R1L1U6R1U1R1U1
 R1U4L3U1L1U1L3U1L3U1L4U2R1U3L1U1L1U1L4U1
 L1U1L1U1L1U2L1D1L2U1L3D2L2U4R1L1D1L3D1L1
 D2L2"
 240 DRAW A\$
 250 DRAW B\$
 260 A\$="":B\$=""
 270 A\$="BM48,55L2U3R1U1R2D2L1D2BM44,83R2
 D1R3D1L1R2D1L1R3E1G1F1R2D1R1LL4BM71,58R2
 D1U2R1U1D2R1D1BM74,44U2L2R1D1L1D1U1L1U1L
 3U1R5U1R2D1R1U1R1L1U3L1U1L1U1L2D2R1D2L1R
 3U1L1U1H2L1U1D1L4R1U1D2R2"
 280 B\$="BM57,36U1R1L6R2U1R4U1D1L2U1L3BM5
 0,33U1R1U1R2U1D1R1BM55,29R3D1L1R2D1L1R3U
 1D1R1BM64,31R1U1R1BM61,34U1R2BM74,32L5U2
 D1R1E1R4D1U2L1R2U1R1D2R1U2R1D1U1R2U2L6R1
 0L2D1L2D1F2R1U1R1U1R2U1R3"
 290 C\$="D1R2U1R1D1R2U1R2U1R7L4D2R3D1R2U1
 R2L2D2L1D3L1D2U1L2D3U2L2D3R1L2D1L4D1L1D1
 L1D1U1L3D1L1D1L1D2L1D1L1D1U1L2U3L1R1U3R1
 U1R2U2L1R1U2L1R1U2L1U2L4BM100,43R1U1D2R1
 U1R1U1D2R1U2R1D2U1R1"
 300 DRAW A\$
 310 DRAW B\$

```

320 DRAW C$
330 A$="":B$="":C$=""
340 A$="BM137,77U1L8U1L1D2L3U1L2U1L3U3R1
L1U1L5D1L4D1L2U1L2D1L1D1L1D1L1D2L1D1L1D1
L1D1L1D2L1D2L1D1L1R1D6L1D2R1D1R1D2R1D2R1
D1R2D1R3U1R2D1R2U1R6D1R1D1R2D6L1R2D2R1D1
R1D5R1D3L1D2L1D2R1D3R1D3R1D5"
350 B$="R1D4R1D1R1U1R4U1R2U1R1U1R1U2R1U3
R2U2R1L1U3R1U2R2U1R2U6L1U5R1U2R1U1R1U1R1
U2R2U1R1U1R1U1R1U2R1U3R1U1L3D1L4U1L1U2L1
U1L1U1L1U2L1U4L1U1L1U2L1U2L1U2R1U2R1U1R1
U4L6U1L1U1L1R1U1R2U1R2U1R2D1R4"
360 C$="U3L1U1L1U1R1U2L3D1L1D2U1L1U2L2D2
L2D2L1D3U1R1L4D4U2L2U3L1U1L1U1L1U1L2D2F3
D1G2E2U1L2H2U1L1U1L2D1L3D1L1D1L2D3L1D1L6
R1U2L1U1R1U3L1R3D1R3U1R1U4L2R2U1R2U1R1U1
H1L1D1U3L1U3H1G1D1R1D1R1D2"
370 DRAW A$
380 DRAW B$
390 DRAW C$
400 A$="":B$="":C$=""
410 A$="L1D1L1R1D2L1R1U2R3F1R1U2D1R2U1R2
U3E1G1D2R1E1G1D1R4U1R1U1D1R2U3D1R2U2R2U1
D1R2U2L1U1L3D1L2U2R1U1R1U2L1U1L2D2L1D1L1
D3R1L2D2R1L1D2U1L1U1L1U1L4R1D1U3L1R1U1R2
U1R1U1R1U1R1U1R1U1R1U1R1U1R3U1D1R2"
420 B$="U1D1R2D1R3D2R1U1R1L5D2R1D1R1D1R2
U1D1R2U2R2U3L1R1D2R2U1R3U1D1R3U1D1R2U2D1
R3U2L1U2R1D1R1D3R2L2D2R1D1R2U2R1U1D1R2U2
L1U1L3U1E1F1R1D1R2U3L2R2U2R5U1R1U1R1D1R4
D2L1D2R2U1R1U1R4D1R3U1R3D1R1D1R2"
430 C$="U1H1F1R3U1R1U1D1R2U1R1D1R2D1R2U1
R2D1R2D1R2U1R5U1R5D1R3D1R2U1R2L1D2R1L1U1
L2D1L2D2R3L1D3L2D1U1L2D3R2D1R1D2R1D3U1L1
U1L1U1L1U1L1U1L1R1U1F2H2U4L1U1L2D2U1L2D3
R1L6D1L1D2L1D1L1D2R2D1R2U1D1"

```

```

440 DRAW A$
450 DRAW B$
460 DRAW C$
470 A$="":B$="":C$=""
480 A$="R1D1R1D1R1D2R1D6L2D3L1D3R2D1R1D1
L1D1U1L1U2L1U2L3D1U2L3D2L1D2R2D1R2U1R1L1
D1L1D2R1D1R1D1R1D2R1D3L1D2L1D2L2D1L3D1U1
L3D5R1D1R1D2R1D3L1D1L1D2U3L2U2L1U2L3D2R1
D1L1D3R1D1R1D3FF2D1R1U1H1U2H5U5"
490 B$="L1U3L2D1L1D1U4L2U2L3D1L2D1L1D1L1
D1L1D1L1D1L1D4L1D3F1R1D2U1L1U1H1L1D1U1L1
U2L1U3L1U2L1U6L2D2U1L2R2U2L2U1L2U1L7U1L4
U1L1U1L1U1L3D3R1D1R1D1R1D1R1D1R2U1R1U1D1
R1D1R1D1R1L1D1L1D2"
500 C$="L1D1L1D1L1D1L2D1L2D1L3U3L1U2L2U2
L1U3L1U1L1U2L1U3L2BM146,61D3R2D2R1D1R1D1
L1D3R5U3L1U1R1U2L2D1U3L1R1U3L4D1L2BM155,
61R1U1R2D1R1D3L3U1L1U2"
510 DRAW A$
520 DRAW B$
530 DRAW C$
540 A$="":B$="":C$=""
550 A$="BM210,50F3D2R1L1D1F1BM215,59R2D1
R2L1D1L1D1L1R1F1D1L1R1D1R1D1L1D1R1D1R1L3
D1R2L1D1U1L4D1BM210,80D3U1L1U1BM211,87D1
R1D1L2R1D1R1D1R1D1R1F1D1BM211,95G2BM212,
93F1BM214,96F1R2U1D2L3R1D1R1"
560 B$="BM191,100D1R1D1R1D1R1D1R1D1R1L2R
1D1R2F1L3F1R2F1E1G1L3F1R2F1L2BM196,100F1
U1H1BM202,105D1R1D2R1D1R3D1U1R1U2R1U2R1L
1U4R1U1L1U1D1L1D1L1D2L1D1L1D1L2U1BM216,1
05L4D2R2L2D1L1D3U2R2D2"
570 C$="BM199,113R6D1L1R2BM209,115E1BM21
1,116E1R1BM215,116E1BM222,107R1D2R1D1R2D
1R1D1R1D3R2D1U1R1U1R2D1R1D1R1D1R1H3U1R1L

```

```

101L101L101L201L101D1L1D1"
580 DRAW A$
590 DRAW B$
600 DRAW C$
610 A$="":B$="":C$=""
620 A$="BM230,118D5R1D4R1D5R1D2R1L1D3L1D
2L1D1L1D1L1D1L1D1L1D2L2D2L3D1L1U1L2U1L1U
3R1U2L2D1G1U3L1U1L5D1L3D1L5D1H1U1R1U1R1U
6R1U2L1R1U2R2U1R3U1R3U2R2U1R1U2R1D1R3U1R
1U1R2U1H1F1R3L1D1L1D2"
630 B$="R1D1R1D1R1D1R2U2R1U4BM221,150D1L
1D1L1U2R2BM230,r
640 C$="BM109,54L1U1D2L1D1U2BM122,30R3D2
U1L2R3F1R1BM127,29R1D1U1R1BM146,34R1U1R3
U1R1BM145,37R1U1D1R1"
650 DRAW A$
660 DRAW B$
670 DRAW C$
680 PSET(25,48),15
690 PSET(26,47),15
700 PSET(27,44),15
710 PSET(25,51),15
720 PSET(25,52),15
730 PSET(60,36),15
740 PSET(70,39),15
750 PSET(68,38),15
760 PSET(67,28),15
770 PSET(70,28),15
780 PSET(218,107),15
790 PSET(119,68),15
800 PSET(218,106),15
810 PSET(119,67),15
820 PSET(201,87),15
830 PSET(123,66),15
840 PSET(201,86),15
850 PSET(122,65),15

```

860 PSET(202,86),15
870 PSET(131,72),15
880 PSET(213,72),15
890 PSET(132,72),15
900 PSET(214,72),15
910 PSET(141,28),15
920 PSET(214,73),15
930 PSET(143,29),15
940 PSET(216,71),15
950 PSET(146,28),15
960 PSET(219,58),15
970 PSET(144,39),15
980 PSET(226,38),15
990 PSET(150,38),15
1000 PSET(227,38),15
1010 PSET(217,110),15
1020 PSET(211,32),15
1030 PSET(219,110),15
1040 PSET(193,31),15
1050 PSET(220,110),15
1060 PSET(192,31),15
1070 PSET(236,112),15
1080 PSET(190,31),15
1090 PSET(237,112),15
1100 PSET(188,31),15
1110 PSET(238,112),15
1120 PSET(188,32),15
1130 PSET(238,110),15
1140 PSET(187,31),15
1150 PSET(239,110),15
1160 PSET(167,30),15
1170 PSET(165,30),15
1180 PSET(165,29),15
1190 PSET(164,29),15
1200 PSET(164,30),15

```

1210 PSET(163,30),15
1220 PSET(162,28),15
1230 PSET(161,29),15
1240 PSET(48,88),15
1250 PSET(240,111),15
1260 PSET(49,88),15
1270 PSET(58,88),15
1280 PSET(50,81),15
1290 PSET(69,60),15
1300 PSET(69,57),15
1310 PAINT(30,60),15
1320 PAINT(60,115),15
1330 PAINT(95,35),15
1340 PAINT(130,95),15
1350 PAINT(170,65),15:PAINT(110,68),15:PAINT(123,48),15:PAINT(206,105),15
1360 PAINT(230,113),15:PAINT(225,135),15
1370 OPEN"GRP:"AS1
1380 A$="MSX-COMPUTERS"
1390 B$="WORLDWIDE"
1400 FORX=80TO81STEP.5
1410 PRESET(X,150),14:PRINT#1,A$
1420 PRESET(X+15,160),14:PRINT#1,B$
1430 NEXT
1440 GOTO 1440

```